

## **REMARKS**

Claims 1-20 are pending. Of those, claims 1, 9 and 15 are independent.

### **Claim Rejection under 35 U.S.C. §112**

Claims 5 and 6 are rejected under 35 U.S.C. 112, second paragraph, as having insufficient antecedent basis. Applicant appreciates the Examiner's attentiveness in calling out typographical errors in claims 5 and 6, which have now been corrected. Withdrawal of this rejection accordingly is requested.

### **Claims Rejections under 35 U.S.C. §102**

Claims 1-20 are rejected under 35 U.S.C. §102(e) as being anticipated by published U.S. Patent Application (pubApp) Publication No. 2002/0133507 to Holenstein et al. (the '507 pubApp). Applicant traverses.

The '507 pubApp (again, published U.S. patent application) is directed to a database replication system that replicates blocks of transaction steps or operations (that have been done to a source database) upon a target database. The emphasis of the '507 pubApp is switching the replication from a synchronous mode to an asynchronous mode, and then back to a synchronous mode, based on detection of selected events.

Replication can be done regardless of whether database (DB) changes are processed in parallel (multi-threaded processing) or serially. As such, the replication system taught by the '507 pubApp is applicable to multi-threading or non-parallel (or, in other words, serialized) database (DB) change processing. Paragraph [0153] of the '507 pubApp describes such flexibility as follows:

**[0153]** For reasons of clarity, the descriptions of the present invention describe the application and replication engine as processing one transaction<sup>[1]</sup> at a time [i.e., non-parallel or serialized], whereas in a typical implementation these components would be "multi-threaded", that is, able to process many transactions simultaneously.

---

<sup>1</sup> The '507 pubApp refers to DB changes as transactions, e.g., lines 4-6 of paragraph [00003] ("Transaction I/O (e.g., inserts, updates, and deletes) applied to one database are applied to the other database and vice-versa.").

Only three other paragraphs in the '507 pubApp make any mention of the root word "thread" or a variant thereof. Those paragraphs are [0027], [0028] and [0334], relevant portions of which are reprinted below (underlined emphasis added) for convenience.

**[0027]** [T]he consumer described herein can process multi-threaded (i.e., overlapping) transactions, ... .

**[0028]** Transaction Receiver--device or object which receives transactions sent by a transaction transmitter for posting to a database. In accordance with the present invention, transaction receivers typically unblock the transaction operations or steps as they are received and apply them into the database. Depending on the nature of the transaction operations or steps, they may be applied in parallel or serially, and the transaction profile may be serialized or multi-threaded (that is, one transaction may be replayed at a time, the transactional order may be altered, and/or the transactions may be replayed in the "simultaneous, intermixed" nature that they occurred in the source database). In one embodiment of the present invention, the transaction receiver is identical to the consumer. In other embodiments, the transaction receiver performs some, but not all, of the functions of the consumer. In a bidirectional database replication scheme, each of the two databases has an associated transaction receiver.

**[0334]** The path that tokens take to arrive in the target can be via many routes. The preferred embodiment of the present invention sends them via the audit trail, interspersed at the appropriate point with transaction steps or operations. These tokens can be "piggybacked" onto the last transaction step or operation for their transaction, as well as onto a transaction step or operation for any other transaction. Piggybacking is one preferred scheme in extensive multi-threaded transaction processing environments.

Only the four noted paragraphs in the '507 pubApp mention the root word "thread" or a variant thereof because, again, the '507 pubApp is concerned with the operating mode of the system during replication, namely whether to switch from the more desired synchronous replication mode to the less desired asynchronous replication mode and vice-versa. Such switching is done whether the underlying system processes DB changes in parallel (multi-threaded processing) or serially. In other words, the '508 pubApp is not concerned with the states of processing threads.

The Examiner has asserted that claim 1 is anticipated by paragraph [0025], line 2, and paragraph [0028], lines 5-7 of the '507 pubApp. Paragraph [0028] was reprinted above. For convenience, paragraph [0025] is reprinted as follows.

**[0025]** Collector--an object or process that reads an audit trail, transaction log file, database change queue or similar structure of a first database, extracts information about specified changes to the first database (e.g., insertions, updates, deletions), and passes that information to the consumer object or process defined below. In Shadowbase.RTM. (a commercially available product made by ITI, Inc., Paoli, Pa.) executing on a COMPAQ NSK (Tandem) source, the collector reads TMF or TM/MP audit trails. In a bidirectional database replication scheme, each of the two databases has an associated collector. The extractor process shown in FIG. 1 of U.S. Pat. No. 5,745,753 (Mosher, Jr.) assigned to Tandem Computers, Inc is similar in operation to the collector.

Among other things, claim 1 recites starting a non-active service thread based conditionally upon information regarding other threads. Where in paragraphs [0025] and [0028] does the '507 pubApp suggest, much less disclose, starting a non-active thread in general? Where in paragraphs [0025] and [0028] does the '507 pubApp suggest, much less disclose, that starting a non-active thread is conditioned upon information regarding other threads?

The Examiner also has relied upon paragraph [0258] of the '507 pubApp. For convenience, paragraph [0258] is reprinted (underlined emphasis added) as follows.

**[0258]** One preferred embodiment of the present invention provides a queue inspection scheme for determining if a synchronous mode is properly functioning. This scheme is illustrated with an example having an originating node with a source database and another node having a target database. Each node has a replication engine and a queue of transactions that were posted to the database at the respective node. The replication engine at each node synchronizes the database at the originating node to the target database at the other node by sending the transactions in the queue to the target database. If the queue at the originating node is not draining, or is draining "too slowly" (i.e., replication latency is above a threshold) then it is presumed that synchronization between the source database at the originating node and the target database at the other node cannot be ensured. The system then reverts to an asynchronous replication mode. In one preferred scheme, the queue of transactions are developed from audit trail entries at the respective node.

Perhaps the Examiner has overestimated the meaning of the term “threshold” in paragraph [0258] of the '507 pubApp. There, the threshold is used to determine if replication should switch from the synchronous mode to the asynchronous mode. Paragraph [0258] is unrelated to starting a thread.

Therefore, a distinction of claim 1 over the '507 pubApp is starting a non-active service thread based conditionally upon information regarding other threads. The noted distinction of claim 1 is shared by claims 2-8, which depend at least indirectly from claim 1, respectively.

Independent claims 9 and 15 recite features similar to claim 1, which similarly represent distinctions over the '507 pubApp. The distinctions of claim 9 and 15 are shared by claims 10-14 and 16-20, which depend at least indirectly from claims 9 and 15, respectively.

In view of the foregoing discussion, the §102(e) rejection of claims 1-20 over the '507 pubApp is improper and Applicant requests that it be withdrawn.

## **CONCLUSION**

The issues in the case are considered to be resolved. Accordingly, Applicant requests a Notice of Allowability.

**<REMAINDER OF PAGE INTENTIONALLY LEFT BLANK>**

Person to Contact

In the event that any matters remain at issue in the application, the Examiner is invited to contact the undersigned at (703) 668-8000 in the Northern Virginia area, for the purpose of a telephonic interview.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies, to charge payment or credit any overpayment to Deposit Account No. 08-2025 for any additional fees under 37 C.F.R. §§1.16 or 1.17; particularly, extension of time fees.

Respectfully submitted,

Terry Robison

By 

Thomas S. Auchterlonie,  
Reg. No. 37,275

P.O. Box 8910  
Reston, VA 20195  
(703) 668-8000

TSA/ewd:tta